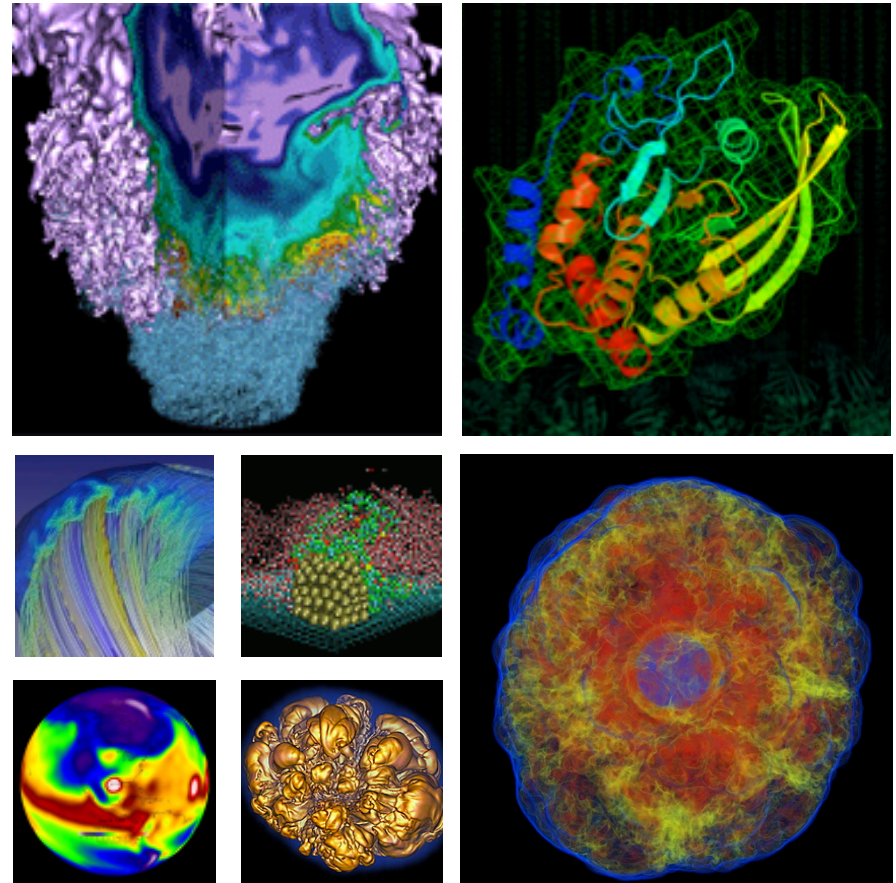


Vtune



T. Koskela
NERSC, LBNL, Berkeley, CA

November 4, 2016

- **Performance analysis tool that enables you to find serial and parallel code bottlenecks and speed execution**
- **Collects samples from your program during run time and prepares various reports on performance**
- **Supports Xeon and Xeon Phi processors**
- **Provides both command line interface ‘amplxe-cl’ and a gui ‘amplxe-gui’**
 - Recommended to use the command line interface in batch or interactive mode on NERSC systems for data collection and the gui over NX for data analysis.

Practical guide



1. Compile your code with ***-g***
 2. In interactive mode add ***--vtune*** to *salloc*, in batch mode add ***#SBATCH -vtune*** to batch script (still necessary on Cori ?)
 3. ***module load vtune***
 4. to run from command line
amplxe-cl -collect <action> -r <result-dir>
 1. on KNL nodes the finalization step is slow, it is recommended to use the ***-no-auto-finalize*** flag in *amplxe-cl* and run *finalize* on the login node
amplxe-cl -finalize -r <result-dir> -search-dir=<path-to-executable> -source-search-dir=<path-to-source>
 5. for list of available actions
amplxe-cl -help collect
for options related to an action
amplxe-cl -help collect <action>
 6. To view results use ***amplxe-gui***
- See <http://www.nersc.gov/users/software/performance-and-debugging-tools/vtune> for more details and example run scripts

General Exploration Viewpoint



- **Note: Each action contains several viewpoints. These can be changed by clicking on change at the top of the window**
- **Summary page: Global Metrics**
 - Cycles Per Instruction (CPI), Investigate if above 4.0
 - Front End Bound: Sign of poor code layout
 - Back End Bound: Sign of memory latency and/or cache misses
- **Bottom-up page: Breakdown by functions**
 - Sort by clockticks to find time-consuming functions
 - Sort by L1 hit rate, L2 hit rate to find bad cache reuse
 - Zoom/Filter in the plot window on the bottom to view specific regions in the execution and/or specific ranks/threads
 - Right-click on rows to view metrics by source line

Other Useful Viewpoints in Gen-Exp



- **Hardware Events**

- Shows the actual counted samples (sample count) and the derived hardware event counts. Useful to determine if sampling rate is appropriate. (For example in the xgc data, L2 miss sample counts are really low)
- Sample count/Event count tabs show the information by function

- **HPC Performance Characterization**

- CPU and FPU utilization (Note that the xgc data has been collected from 1 of 4 mpi processes)
- Bottom Up tab shows the data by function

Advanced Hotspots Viewpoint



- **Summary page:**
 - Identify hotspots based on CPU time usage
 - Useful for debugging, e.g. OpenMP overhead
 - Use `-knob analyze-openmp`
- **Bottom-up page:**
 - Zoom on plot and enable hardware event sample to get another measure of the sampling rate
 - Hover over samples to see individual sample data
 - To collect call stacks, enable call stack collection
 - Use `-knob collection-detail=stack-sampling`
 - Note: has high overhead and may skew profiling results (ie. show high spin time etc.)

Memory Access Viewpoint



- **Summary page:**
 - LLC miss count
 - Bandwidth utilization histogram: DRAM, MCDRAM
- **Bottom-up page:**
 - Plot on the top shows DRAM, MCDRAM bandwidths as function of time during execution
 - Can zoom in/out and filter
 - LLC miss counts per function
 - Use `-knob analyze-mem-objects=true` to analyze data structures



Backup Slides